UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/686,746	10/16/2003	Takeshi Tanaka	82478-1300	8702

21611 7590 07/24/2006

SNELL & WILMER LLP
600 ANTON BOULEVARD
SUITE 1400
COSTA MESA, CA 92626

EXAMINER
LAI, VINCENT

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 07/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/686,746	TANAKA ET AL.	
	Examiner	Art Unit	
	Vincent Lai	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 May 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Fritz Fleming

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
7/19/2006

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Acknowledgment is made of amendments made to the title and claims.

2. Objections to the title and claims have been withdrawn in light of the amendments submitted on 17 May 2006.

Priority

3. Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-7, and 10-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knight, Jr. (U.S. Patent # 4,825,360), herein known as Knight, Jr. in view of Emma et al (U.S. Patent # 5,297,281), herein referred to as Emma et al.

As per claim 1, Knight Jr. teaches a parallel execution processor (See figure 1) comprising: a plurality of processing elements (Processor 1a, 1b, 1c, see figure 1); an obtaining unit (Compiler 9, see figure 1) operable to obtain an instruction sequence including one or more instructions (See column 4, lines 25-27: The compiler sequences all instructions into blocks); a group forming unit (Compiler 9, see figure 1) operable to form the processing elements (See column 7, lines 49-52: The compiler creates blocks) into groups; and an execution controlling unit (Compiler 9, see figure 1) operable to assign part or all of the instructions included in the decoded instruction sequence to the groups (See column 7, lines 65-68: The compiler does the assigning of instructions to groups), so that one group receives one instruction (See column 7, lines 65-68: One block is given to one processor), and control the processing elements so that (i) the instructions received by the groups are executed in parallel (See column 7, lines 65-68: As part of a parallel processor), and (ii) in each group, all processing elements in the group are employed in parallel for the execution of the received instruction (See column 7, lines 65-68: Blocks are executed in parallel).

Knight, Jr. does not teach the use of group number information indicating how many groups the processing elements should be formed into or a decoding unit.

Emma et al does teach the use of group number information (Group number (GN), see column 7, lines 25-30) and a decoding unit (Decode stage, see figure 2B, and column 8, lines 48-51) operable to decode the obtained instruction sequence (See column 8, lines 48-51: Instructions are decoded).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. to the use of group number information, and a decoding unit.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. by the teachings of Emma et al. Although Knight, Jr. teaches the use of a block pointer to indicate a block (or group) of instructions and a block counter for counting the number of blocks (and tracking the PC of each block), group number information would be able to advantageous since it would compact two features into one, meaning less hardware is used and less information needs to be tracked at once.

Many of the teachings of Emma et al can be viewed as improvements to the design of Knight, Jr. by simplifying tasks (and thereby meaning less hardware and less data handled, which leads to less chance of errors), and/or improving the ease of identifying data (and thereby reducing chance of errors). These serve as a motivation for combining Emma et al with Knight, Jr. for the remainder of the claims below.

As per claim 2, Knight, Jr. teaches an instruction sequence includes as many instructions as the number of groups (See column 4, lines 25-27: The compiler sequences all instructions into blocks (or groups) and there is no block of no instructions)

Knight, Jr. does not teach wherein the instruction sequence includes as many instructions as the number of groups indicated by the piece of group number information.

Emma et al does teach the use of group number information (Group number (GN), see column 7, lines 25-30), and also include means for indicating the end of instructions in an instruction sequence of a group, meaning there will not be more groups than possible number of instructions (See column 11, lines 29-37: A new group is created when necessary, with and END in the instruction queue to account for the end of the group).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to use group number information that indicates end of instruction sequences.

As per claim 3, Knight, Jr. teaches the ability to split the instruction into two groups (See column 7, lines 49-52: There is limit to how many groups are to be formed), each group processed on its own processor (See column 7, lines 65-68: One block is given to one processor).

Knight, Jr. does not teach wherein a piece of the group number information is indicative of how many groups are to be formed.

Emma et al teaches wherein the number of groups indicated by the piece of group number information is either one or two (See column 7, lines 50-65: The group selection circuitry deals with a 1-bit flag which is indicative of how many processors are

to be used), when the number of groups indicated is one, the group forming unit forms all of the processing elements into one group (See column 7, lines 50-57: When the group selection circuitry has a 0), and when the number of groups indicated is two, the group forming unit forms all of the processing elements into two groups so that the two groups contain an equal number of processing elements (See column 7, lines 57-65: When the group selection circuitry has a 1).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to have a piece of the group number information is indicative of how many groups are to be formed.

As per claim 4, Knight, Jr. teaches the use of a register (See column 7, lines 55-60: Each processor has registers), and an instruction sequence that includes a first instruction and a second instruction (See column 4, lines 25-27: The compiler sequences all instructions into blocks).

Knight, Jr. does not teach the use of register files and register files such that they are arranged in a manner so that they correspond to only one processor.

Emma et al does teach a register that includes a plurality of register files (See figure 1, and column 10, lines 47-52: Many registers 130 are available) each of which corresponds to a different one of the processing elements (See figure 3B and column 14, lines 31-32: Registers are shared, but is assigned a processor when it is read by a processor), wherein the register files are arranged in the register so that first-group register files and second-group register files alternate (See figure 2H, and column 7, line

65- column 8, line 16: There is a flip-flop 218 which alternates the group dispatcher 120 to work on the main pipeline and then the auxiliary pipeline, which when applied to Knight, Jr. can just be between the first and the second processor), (i) the first-group register files each storing therein a piece of data to be processed when the first instruction is executed (See column 7, lines 14-16: Instructions are stored) and (ii) the second-group register files each storing therein a piece of data to be processed when the second instruction is executed (See column 7, lines 14-16: Instructions are stored), when the number of groups indicated is two (See column 7, lines 57-65: When the group selection circuitry has a 1), the group forming unit forms the processing elements corresponding to the first-group register files into one of the two groups, and the processing elements corresponding to the second-group register files into the other group (See column 7, lines 40-44: Group dispatcher chooses which registers to assign to each processor), and each of the processing elements obtains the piece of data to be processed from the corresponding register file (See column 7, lines 17-20 and 40-44: Registers are used as an associative memory and is assigned to each processor and thus the processor will get its data from the register).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to have register files arranged in a manner so that they correspond to only one processor.

As per claim 5, Knight, Jr. teaches the use of a register (See column 7, lines 55-60: Each processor has registers).

Knight, Jr. does not teach a plurality of pairings of register files and its use.

Emma et al teaches wherein the register files are formed into a plurality of pairs (See column 7, lines 8-10: The registers have both BA and TA addresses), keeping an order in which the register files are arranged in the register (See column 5, lines 24-27: Instructions are arranged for the processors), each of the instructions includes a piece of selection information (See column 7, lines 50-65: The group selection circuitry acts according to information in instruction) indicating which piece of data each processing element should obtain, selecting out of (a) the piece of data stored in the corresponding register file (See column 7, lines 8-10: Such as the BA register) and (b) the piece of data stored in a register file with which the corresponding register file is paired (See column 7, lines 8-10: Such as the TA register), and each of the processing elements obtains the piece of data to be processed from the register file indicated in each piece of selection information (See column 7, lines 40-44: The group dispatcher selectively applies the TA, BA fields).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to have register files arranged in a manner so that they correspond to only one processor.

As per claim 6, Knight, Jr. teaches an execution controlling unit (Compiler 9, see figure 1).

Knight, Jr. does not teach what is to occur when the number of group indicated is two or the execution controlling unit having various other units to make use of group number information.

Emma et al teaches wherein when the number of groups indicated is two (See column 7, lines 57-65: When the group selection circuitry has a 1), the execution controlling unit includes: a storing unit (Instruction count registers 220, 230, see figure 2A) that stores therein a plurality of combination options based on which of the processing elements should belong to each of the two groups (See figure 2A, and column 7, lines 40-49: The main and auxiliary instruction count registers 220, 230, respectively are the storing units assigned to each group), the combination options being prepared for each of a plurality of grouping procedures; a grouping information obtaining unit (Group selection circuitry 122, see figure 2A) operable to obtain a piece of grouping information indicating which one of the grouping procedures should be used (See column 7, lines 50-65: The group selection circuitry acts according to information in instruction); and a selecting unit (Group dispatcher 120, see figure 2A) operable to select one of the combination options according to the obtained piece of grouping information (See column 7, lines 40-49: The group dispatcher does the selection).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to include a description of what is to occur when the number of group indicated is two and the various other units to make use of group number information (which can be put into the compiler of Knight, Jr.).

As per claim 7, Knight, Jr. teaches the grouping of processing elements (See column 7, lines 49-52: The compiler creates blocks) and does assign which group goes with which processing element (See column 7, lines 65-68: Blocks are assigned).

Knight, Jr. does not teach group number information and its relation to grouping processing elements.

Emma et al teaches wherein when the number of groups indicated is two (See column 7, lines 57-65: When the group selection circuitry has a 1), the execution controlling unit includes: a grouping information obtaining unit (Group selection circuitry 122, see figure 2A) operable to obtain a piece of grouping information indicating to which one of the two groups each of the processing elements should belong (See column 7, lines 50-65: The group selection circuitry acts according to information in instruction); and a grouping unit operable to form the processing elements into the two groups according to the obtained piece of grouping information (See column 7, lines 50-65: The group selection circuitry acts according to what bit it is given).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to include a description of use of group number information.

As per claim 10, Knight, Jr. teaches the plurality of processing elements (Processor 1a, 1b, 1c, see figure 1) and an execution controlling unit (Compiler 9, see figure 1).

Knight, Jr. does not teach the use of a halt operation that stops operations in the plurality of processing elements and its effects on an execution controlling unit.

Emma et al teaches wherein when the number of groups indicated by the piece of group number information is two or larger (See column 7, lines 50-65: The group selection circuitry can work up to four groups), the obtaining unit obtains an instruction that instructs that processing elements included in some of the groups should halt operation (See column 9, lines 29-42: Error signals indicate when operations should halt).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to include a halt operation in the event of an error that may occur.

As per claim 11, Knight, Jr. teaches a parallel execution processor (See figure 1) comprising: a plurality of processing elements (Processor 1a, 1b, 1c, see figure 1); a register (See column 7, lines 55-60: Each processor has registers), an obtaining unit (Compiler 9, see figure 1) operable to obtain an instruction sequence that includes the first instruction and the second instruction (See column 4, lines 25-27: The compiler sequences all instructions into blocks); a group forming unit (Compiler 9, see figure 1) operable to form the processing elements (See column 7, lines 49-52: The compiler creates blocks) into groups; and an execution controlling unit (Compiler 9, see figure 1) operable to assign (i) the first instruction to the processing elements corresponding to the first-group register files and (ii) the second instruction to the processing elements

corresponding to the second-group register files (See column 7, lines 65-68: The compiler does the assigning of instructions to groups), and control the processing elements so that (i) the first and second instructions are executed in parallel (See column 7, lines 65-68: As part of a parallel processor), (ii) the processing elements executing the first instruction are employed in parallel for the execution (See column 7, lines 65-68: Blocks are executed in parallel), and (iii) the processing elements executing the second instruction are employed parallel for the execution (See column 7, lines 65-68: Blocks are executed in parallel).

Knight, Jr. does not teach the use of group number information, a decoder, or a plurality of pairings of register files and its use.

Emma et al teaches the use of group number information (Group number (GN), see column 7, lines 25-30). Emma et al also teaches a decoding unit (Decode stage, see figure 2B, and column 8, lines 48-51) operable to decode the first instruction and the second instruction included in the obtained instruction sequence (See column 8, lines 48-51: Instructions are decoded).

Emma et al further teaches a register that includes a plurality of register files (See figure 1, and column 10, lines 47-52: Many registers 130 are available) each of which corresponds to a different one of the processing elements (See figure 3B and column 14, lines 31-32: Registers are shared, but is assigned a processor when it is read by a processor), the register files being arranged in the register so that first-group register files and second-group register files are positioned according to a predetermined rule (See column 5, lines 24-27: Instructions are arranged for the

processors), (i) the first-group register files each storing therein a piece of data to be processed when a first instruction is executed (See column 7, lines 8-10: Such as the BA register) and (ii) the second-group register files each storing therein a piece of data to be processed when a second instruction is executed (See column 7, lines 8-10: Such as the TA register).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to use group number information, have a decoder, and have a plurality of pairings of register files.

As per claim 12, Knight, Jr. teaches a register (See column 7, lines 55-60: Each processor has registers).

Knight, Jr. does not teach a plurality of register files and its arrangement.

Emma et al teaches wherein the register files are arranged in the register so that first-group register files and second-group register files alternate (See figure 2H, and column 7, line 65- column 8, line 16: There is a flip-flop 218 which alternates the group dispatcher 120 to work on the main pipeline and then the auxiliary pipeline, which when applied to Knight, Jr. can just be between the first and the second processor).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to include a plurality of register files.

As per claim 13, Knight, Jr. teaches a register (See column 7, lines 55-60: Each processor has registers).

Knight, Jr. does not teach a plurality of register files and its correspondence to processing elements.

Emma et al teaches wherein the register that includes a plurality of register files (See figure 1, and column 10, lines 47-52: Many registers 130 are available) each of which corresponds to a different one of the processing elements (See figure 3B and column 14, lines 31-32: Registers are shared, but is assigned a processor when it is read by a processor), each of the instructions includes a piece of selection information (See column 7, lines 50-65: The group selection circuitry acts according to information in instruction) indicating which piece of data each processing element should obtain, selecting out of (a) the piece of data stored in the corresponding register file (See column 7, lines 8-10: Such as the BA register) and (b) the piece of data stored in a register file with which the corresponding register file is paired (See column 7, lines 8-10: Such as the TA register), and each of the processing elements obtains the piece of data to be processed from the register file indicated in each piece of selection information (See column 7, lines 40-44: The group dispatcher selectively applies the TA, BA fields).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to have register files arranged in a manner so that they correspond to only one processor.

As per claim 14, Knight, Jr. teaches an instruction assigning method for assigning instructions to a plurality of processing elements (Processor 1a, 1b, 1c, see figure 1), comprising an obtaining step of obtaining an instruction sequence including one or more instructions (See column 4, lines 25-27); and an execution controlling step of assigning part or all of the instructions included in the decoded instruction sequence to the groups, (See column 7, lines 65-68: The compiler does the assigning of instructions to groups) so that one group receives one instruction (See column 7, lines 65-68: One block is given to one processor), and controlling the processing elements so that (i) the instructions received by the groups are executed in parallel (See column 7, lines 65-68: As part of a parallel processor), and (ii) in each group, all processing elements in the group are employed in parallel for the execution of the received instruction (See column 7, lines 65-68: Blocks are executed in parallel).

Knight, Jr. does not teach a method to obtain group number information, and a decoding step.

Emma et al teaches use of group number information (Group number (GN), see column 7, lines 25-30) and a decoding step of decoding the obtained instruction sequence; a group forming step of forming the processing elements into as many groups as indicated by the piece of group number information (See column 8, lines 48-51: Instructions are decoded);

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Knight, Jr. with Emma et al to include a the use of group number information and a decoding step.

As per claim 15, Knight, Jr. teaches an instruction assigning method for assigning a first instruction and a second instruction to a plurality of processing elements (See column 7, lines 65-68: The compiler assigns instructions in blocks to individual processors), the instruction assigning method comprising: a storing step of (i) reading as many pieces of data as the number of processing elements (See column 7, lines 65-68: Compiler reads all instructions into blocks), from a memory (Main memory 6 and 7, see figure 1) in which (a) pieces of data to be processed when a first instruction is executed and (b) piece of data to be processed when a second instruction is executed are arranged in an order according to a predetermined rule (See column 7, lines 65-68: Instructions are placed in order in blocks and are executed by processors) and (ii) storing the pieces of data, without changing the order, into register files each of which corresponds to a different one of the processing elements (See column 7, lines 65-68: Each block has its own processor); an obtaining step of obtaining an instruction sequence that includes the first instruction and the second instruction (See column 7, lines 65-68: The compiler receives all instructions to sequence); and an execution controlling step of assigning (i) the first instruction to the processing elements corresponding to the register files that each store therein the piece of data to be processed when the first instruction is executed and (ii) the second instruction to the processing elements corresponding to the register files that each store therein the piece of data to be processed when the second instruction is executed (See column 7, lines 65-68: The compiler does the assigning of instructions to groups), and controlling the

processing elements so that (i) the first and the second instructions are executed in parallel (See column 7, lines 65-68: As a part of a parallel processor), (ii) the processing elements executing the first instruction are employed in parallel for the execution (See column 7, lines 65-68: Blocks are executed in parallel), and (iii) the processing elements executing the second instruction are employed in parallel for the execution (See column 7, lines 65-68: Blocks are executed in parallel).

Knight, Jr. does not teach a decoding step.

Emma et al teaches a decoding step of decoding the first instruction and the second instruction included in the obtained instruction sequence (See column 8, lines 48-51: Instructions are decoded).

5. Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knight, Jr. (U.S. Patent # 4,825,360), herein known as Knight, Jr. in view of Emma et al (U.S. Patent # 5,297,281), herein known as Emma et al, in further view of Rymarczyk (Coding Guidelines for Pipelined Processors), herein known as Rymarczyk.

As per claim 8, Knight, Jr. teaches the use of a parallel execution processor (See figure 1).

Knight, Jr. does not teach a fetching unit.

Emma et al teaches the use of a fetching unit (Group dispatching circuit 120, see figure 2A) operable to fetch a piece of data which is of a predetermined length and has a format field and a data field (See column 8, lines 51-55: Data is fetched according to signals)

Emma et al does not teach wherein each of the instructions includes an OP code and an operand, a positioning pattern is written in the format field, the positioning pattern being for positioning OP codes and operands in the data field in the piece of data, one or more OP codes and one or more operands are arranged in the data field in an order defined by the positioning pattern written in the format field, the number of groups indicated by the piece of group number information is a number of instructions defined by the positioning pattern, the decoding unit extracts, from the piece of data, the one or more OP codes and the one or more operands, according to the positioning pattern so as to decode the OP codes and the operands of the instructions, and the execution controlling unit assigns, in the defined order, the decoded instructions to the groups.

Rymarczyk teaches the steps to process an instruction, citing the IBM 3033 processor—which is also cited by Emma et al (See column 4, lines 58-60). Rymarczyk teaches the use of machine code, wherein machine code has a positioning pattern is written in the format field, the positioning pattern being for positioning OP codes and operands in the data field in the piece of data, one or more OP codes and one or more operands are arranged in the data field in an order defined by the positioning pattern written in the format field (See page 12, under "Introduction to Pipelined Processors": Machine code has been understood as being a part of all processors in order for them to be of any use) and OP codes are a form of machine code (Operands are an inherent part of op codes).

Rymarczyk also teaches wherein the number of groups indicated by the piece of group number information is a number of instructions defined by the positioning pattern, the decoding unit extracts, from the piece of data, the one or more OP codes and the one or more operands, according to the positioning pattern so as to decode the OP codes and the operands of the instructions, and the execution controlling unit assigns, in the defined order, the decoded instructions to the groups (See pages 12-13, under "Steps in the Processing of an Instruction": These are normal steps in processing an instruction).

Therefore it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Knight, Jr. to include the disclosure of the machine code available to processors to allow ease of programmability.

It would have been obvious to a person ordinary skill in the art at the time of the invention was made to have modified Knight, Jr. with Emma et al by the teachings of Rymarczyk, because the teachings of Rymarczyk are already a part of what is taught of Emma et al, but is not disclosed in Emma et al.

As per claim 9, Knight, Jr. teaches the use of a parallel execution processor (See figure 1).

Knight, Jr. does not teach a fetching unit.

Emma et al teaches the use of a fetching unit (Group dispatching circuit 120, see figure 2A) operable to fetch a piece of data which is of a predetermined length and has

a format field and a data field (See column 8, lines 51-55: Data is fetched according to signals).

Emma et al does not teach a storing unit operable to store therein a predetermined positioning pattern for OP codes and operands, wherein each of the instructions includes an OP code and an operand, one or more OP codes and one or more operands are arranged in the piece of data in an order defined by the predetermined positioning pattern, the number of groups indicated by the piece of group number information is a number of instructions defined by the positioning pattern, the decoding unit extracts, from the piece of data, the one or more OP codes and the one or more operands, according to the positioning pattern so as to decode the OP codes and the operands of the instructions, and the execution controlling unit assigns, in the defined order, the decoded instructions to the groups.

Rymarczyk teaches a storing unit operable to store therein a predetermined positioning pattern for OP codes and operands, wherein each of the instructions includes an OP code and an operand, one or more OP codes and one or more operands are arranged in the piece of data in an order defined by the predetermined positioning pattern, the number of groups indicated by the piece of group number information is a number of instructions defined by the positioning pattern, the decoding unit extracts, from the piece of data, the one or more OP codes and the one or more operands, according to the positioning pattern so as to decode the OP codes and the operands of the instructions, and the execution controlling unit assigns, in the defined order, the decoded instructions to the groups (See pages 12-13, under "Steps in the

Processing of an Instruction": These are normal steps in processing an instruction) in a IBM 3033 processor, which is also cited by Emma et al (See column 4, lines 58-60).

Therefore it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Knight, Jr. to include the disclosure of how data is handled in an IBM 3033 processor.

It would have been obvious to a person ordinary skill in the art at the time of the invention was made to have modified Knight, Jr. with Emma et al by the teachings of Rymarczyk, because the teachings of Rymarczyk are already a part of what is taught of Emma et al, but is not disclosed in Emma et al.

Response to Arguments

6. Applicant's arguments filed 17 May 2006 have been fully considered but they are not persuasive.

Applicant's arguments are moot in view of the claims. It is the contention of the examiner that Knight, Jr. in view of Emma et al (and subsequently Knight, Jr. in view of Emma et al in further view of Rymarczyk) cover all previous and amended limitations claimed. Whereas applicant may have potentially pointed out deficiencies in the obviousness rejections in light of the entirety of the invention, the obviousness rejections can still be applied to the claimed limitations when the broadest interpretation of the claims are used.

Using claim 1 as an example, if each processor is itself a group (which is not disallowed by the claims and is the case in Knight, Jr.), then indeed all processors are divided into groups and thus all each group or block of instructions is assigned to a different group of processing elements.

It is of note that figure 3 of Knight, Jr. shows the blocks are processed at the same time, and not at different times as the remarks contend.

The use of Knight, Jr. and Emma in a combination is to provide an actual group number. Knight, Jr. already does similar group tasks but does not explicitly or implicitly teach a group number field. Emma does teach a group number field and is similar in scope to Knight, Jr. Utilizing a group number field would only improve the invention of Knight, Jr. and thus would be an obvious combination.

Using similar reasoning, the above will apply to claims 7, 11, and 14-15.

The rejection utilizing Knight, Jr. in view of Emma et al in further view of Rymarczyk still applies to claims 8-9 in light of the above reasoning.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Vincent Lai whose telephone number is (571) 272-6749. The examiner can normally be reached on M-F 8:00-5:30 (First BiWeek Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz M. Fleming can be reached on (571) 272-4145. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Vincent Lai
Examiner
Art Unit 2181

vl
July 16, 2006

Fritz Fleming

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

7/19/2006